Algebraic analogues to fundamental notions of query and dependency theory

Serge Abiteboul

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.: 954 90 20

Rapports de Recherche

N° 201

ALGEBRAIC ANALOGUES TO FUNDAMENTAL NOTIONS OF QUERY AND DEPENDENCY THEORY

Serge ABITEBOUL

Avril 1983

ALGEBRAIC ANALOGUES TO FUNDAMENTAL NOTIONS OF QUERY AND DEPENDENCY THEORY

by

Serge Abiteboul

Résumé

Nous présentons des analogues algébriques des requêtes conjonctives, des homomporphismes de requêtes et d'une large classe de dépendances ("non-typed embedded implicational dependencies"). Nous introduisons aussi des contreparties algébriques pour des dépendances générant des n-uplets ou des égalités ("tgd, egd"). Nous montrons comment simuler une poursuite impitoyable ("chase") en utilisant un système de règles d'inférences. Finalement, nous étudions plus particulièrement les requêtes typées.

Abstract

We present algebraic analogues to conjunctive queries, query homomorphism and (non-typed) embedded implicational dependencies (NEIDs). We exhibit also algebraic counterparts for tuple and equality generating dependencies. We show how to simulate the chase using a sound and complete set of inference rules. Finally, we consider the typed queries.

Within the field of relational databases, the topics of containment of mappings and dependency implication have attracted particular attention [ASU, Ar, BFH, BMSU, BV, CM, MMS, SU, YP]. Two tools have turned to be essential to solve these problems, namely, homomorphism [CH] and chase [ABU,BV3]. However, both of them deal with calculus oriented definitions of mappings and dependencies. The purpose of the present paper is to exhibit a strong parallelism between calculus and algebraic sides of the two problems. In particular, algebraic counterpart to homomorphism and chase are presented. (Indeed, the chase is simulated by a sound and complete set of inference rules for non-typed embedded implicational dependencies. This generalizes [YP] results obtained in the typed case).

In all the paper, we tried to emphasize the parallel between the calculus and algebraic sides of the problem. For instance, algebraic counterpart to tableau generating dependencies and equality generating dependencies are presented. Furthermore, a strong link is established between variables in tableaux, conjunctive queries or non-typed embedded implicational dependencies and attributes in corresponding algebraic expressions. In this context, a very simple operation, namely, renaming, can be thought as an algebraic counterpart of isomorphism.

The paper is organized as follows. In Section 1, some formalism is presented for tableau, conjunctive queries, **tgds** and **egds**. Four algebraic operators (projection, cartesion product, restriction and renaming), algebraic mappings and algebraic dependencies are introduced. Section 2 presents briefly "bridges" to link some of these concepts. In Section 3, containment between mappings is considered and, in Section 4, implication of dependencies. In Section 5, some particular mappings and dependencies, namely, the typed ones, are considered.

I. - PRELIMINARIES

In this section, we present some well-known concepts of the relational model used throughout the paper. We assume the existence of an infinite set $\mathcal{D}$ of values called the domain. We also assume the existence of an infinite set $\mathcal{S}$ of symbols. So-called attributes and variables will both be taken in this set $\mathcal{S}$. We will keep as much as possible the convention to use A, B, C, ... to denote attributes, $\mathcal{U}, \mathcal{V}, \mathcal{W}$ ... sets of attributes, x, y, z variables and a, b, c constants. We also adopt the convention to denote $\mathcal{U}\mathcal{V}$ the union of $\mathcal{U}$ and $\mathcal{V}$.

Let $\mathcal{U}$ be a finite set of attributes. Then a value tuple is a mapping from $\mathcal{U}$ to $\mathcal{D}$, and a variable tuple is mapping from $\mathcal{U}$ to $\mathcal{S}$. The set of all value tuples over $\mathcal{U}$ is denoted tup($\mathcal{U}$), and the set of all variable tuples over $\mathcal{U}$ denoted v-tup($\mathcal{U}$). A finite set of value tuples over $\mathcal{U}$ is called an instance or a relation. A finite set of variable tuples over $\mathcal{U}$ is called a tableau. The set of all instances over $\mathcal{U}$ is denoted Inst($\mathcal{U}$), and the set of all tableaux over $\mathcal{U}$ denoted Tab($\mathcal{U}$).

For each tableau T over $\mathcal{U}$, Var(T) denotes the set $\{t(A)/t$ in T, A in $\mathcal{U}\}$ of variables appearing in T. Let T be a tableau over $\mathcal{U}$ and I be an instance over $\mathcal{U}$. Then a valuation of T in I is a mapping from Var(T) into $\mathcal{D}$ such that vT $\subseteq$ I where vT = $\{vt/t$ in T$\}$.

Let T be a tableau over $\mathcal{U}$ and t a variable tuple over $\mathcal{V}$ such that Var(t) $\subseteq$ Var(T). Then Q $\equiv$ < T,t > is called a conjunctive query(), $\mathcal{U}$ is denoted α(Q) and $\mathcal{V}$ denoted β(Q). The conjunctive query Q defines a conjunctive query mapping from Inst (α(Q)) into Inst(β(Q)) in the following way :

Q(I) = $\{vt/v$ is a valuation of T in I$\}$

for each I in Inst (α(Q)).

() the tuple t is sometimes called the summary of Q.

We now recall some relational operators.

Definition

Let $\mathcal{U}$, $\mathcal{U}'$ be two finite sets of attributes with $\mathcal{U} \cap \mathcal{U}' = \emptyset$. Let I, I' be two instances over $\mathcal{U}$, $\mathcal{U}'$, respectively. Let $\mathcal{V} \subseteq \mathcal{U}$, $AB \subseteq \mathcal{U}$ and ρ be a one-to-one mapping from $\mathcal{U}$ onto W. Then

(a) the projection() of I on V, denoted $\Pi_V(I)$, is defined by $\Pi_V(I) = \{u[V]/u \text{ in } I\}$;

(b) the restriction of I by A = B, denoted $\sigma_{A=B}(I)$, is defined by $\sigma_{A=B}(I) = \{u/u(A) = u(B)\}$;

(c) the renaming of I by ρ, denoted $\rho(I)$, is defined by $\rho(I) = \{\rho u/u \text{ in } I\}$ where ρu is the tuple over W defined by $\rho u(\rho(A)) = u(A)$ for each A in $\mathcal{U}$;

and

(d) the cartesion product of I and I', denoted I x I', is defined by $I \times I' = \{w \text{ in } Rel(\mathcal{U}\mathcal{U}')/w[\mathcal{U}] \text{ in } I \text{ and } w\ [\mathcal{U}'] \text{ in } I'\}$.

In the present paper, we are concerned with mappings obtained by finite applications of these four operators. To define them, we need the concept of (algebraic) expression.

Definition

(1) for each finite set $\mathcal{U}$ of attributes, $\mathcal{U}$ is an expression, $\alpha(\mathcal{U}) = \mathcal{U}$ and $\beta(\mathcal{U}) = \mathcal{U}$;

() for each tuple u over $\mathcal{U}$ and $V \subseteq \mathcal{U}$, u[V] denotes the restriction of u to V.

(2) Let φ_1, φ_2 be two expressions with $\alpha(\varphi_1) = \alpha(\varphi_2)$ and $\beta(\varphi_1) \cap \beta(\varphi_2) = \emptyset$.
Then $\varphi_1 \times \varphi_2$ is an expression, $\alpha(\varphi_1 \times \varphi_2) = \alpha(\varphi_1)$, $\beta(\varphi_1 \times \varphi_2) = \beta(\varphi_1) \cup \beta(\varphi_2)$;

(3) Let φ be an expression, $V \subseteq \beta(\varphi)$ and $AB \subseteq \beta(\varphi)$.
Then $\pi_V(\varphi)$, $\sigma_{A=B}(\varphi)$ are expressions, $\alpha(\pi_V(\varphi)) = \alpha(\sigma_{A=B}(\varphi)) = \alpha(\varphi)$, $\beta(\pi_V(\varphi)) = V$ and $\beta(\sigma_{A=B}(\varphi)) = \beta(\varphi)$.

(4) Let φ be an expression and ρ a one-to-one mapping over $\beta(\varphi)$.
Then $\rho(\varphi)$ is an expression, $\alpha(\rho(\varphi)) = \alpha(\varphi)$ and $\beta(\rho(\varphi)) = \rho(\beta(\varphi))$.

The associated mappings are defined in the straightforward way ($\mathcal{U}$ is the identity mapping over Inst($\mathcal{U}$)).

We will consider also some (general) dependencies.

Definition

Let φ and Ψ be two expressions with $\alpha(\varphi) = \alpha(\Psi)$ and $\beta(\varphi) = \beta(\Psi)$. Then $\varphi \subseteq \Psi$ is called a dependency over $\alpha(\varphi)$. An instance I over $\alpha(\varphi)$ satisfies $\varphi \subseteq \Psi$ iff $\varphi(I) \subseteq \Psi(I)$.

We assume standard knowledge of the notions of equivalence between sets of dependencies and logical implication (denoted $\models$).

II. - BASIC COUNTERPARTS

Three results linking various concepts presented in the preliminaries are exhibited. These results will be extensively used in the following sections. The first one establishes a correspondance between valuations and some particular expressions. To prove it, we need the following construction.

Construction 1 : From tableau to expression

Let $T = \{t_k \ / \ k \text{ in } K\}$ be a tableau over $\mathcal{U}$. For each x in Var(T), let $\hat{k}(x)$ in K and $\hat{A}(x)$ in $\mathcal{U}$ be such that $x = t_{\hat{k}(x)}(\hat{A}(x))$.

Let $\{A_k$ / k in K, A in $\mathcal{U}\}$ be a set of distinct new attributes. Finally, let

$$\varphi_T \equiv \Pi_{Var(T)} \quad \rho\Sigma \left(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U}) \right),$$

where

(1) for each k, ρ_k is the renaming over $\mathcal{U}$ defined by

$\rho_k(A) = A_k$ for each A in $\mathcal{U}$;

(2) Σ is a finite string of restrictions such that

$\{\sigma$ in $\Sigma\} = \{\sigma_{A_i=B_j}$ / i,j in K, A, B in $\mathcal{U}$ and

$t_i(A) = t_j(B)\}$; and

(3) ρ is the renaming over $\{A_k$ / k in K, A in $\mathcal{U}\}$ defined by $\rho(\hat{A}(x)_{\hat{k}(x)}) = x$ for each x in Var(T) and $\rho(A_k) = A_k$ otherwise. □

Note that $\varphi_T(I)$ is a relation over Var(T). Since a valuation v of T is a mapping from Var(T) into $\mathcal{D}$, v is also a tuple over Var(T). The first result of this section states that the set of valuations of T in I coincides with $\varphi_T(I)$.

Theorem 2.1.

Let T be a tableau over $\mathcal{U}$ and I an instance over $\mathcal{U}$. Then $\varphi_T(I) = \{v$ / v is a valuation of T in I$\}$.

Proof :

Let v be in $\varphi_T(I)$. Then there exists $\{u_k$ in I / k in K$\}$ such that $v = \Pi_{Var(T)}\ \rho\Sigma(\underset{k \text{ in } K}{X} \rho_k(u_k))$. Since v is a mapping from Var(T) into $\mathcal{D}$, to show that v is a valuation of T in I, it suffices to prove that vt_ℓ is in I for each ℓ in K. Let ℓ be in K, A in $\mathcal{U}$ and $x = t_\ell(A)$. Then $vt_\ell(A) = v(x) = u_{\hat{k}(x)}(\hat{A}(x))$. Since $t_\ell(A) = x = t_{\hat{k}(x)}(\hat{A}(x))$, $\sigma_{A_\ell=\hat{A}(x)_{\hat{k}(x)}}$ is in Σ. Thus $u_{\hat{k}(x)}(\hat{A}(x)) = u_\ell(A)$. Therefore $vt_\ell(A) = u_\ell(A)$.

Hence $vt_\ell(B) = u_\ell(B)$ for each B in $\mathcal{U}$, i.e., $vt_\ell = u_\ell$. Consequently, vt_k is in I for each k in K, that is, v is a valuation of T in I.

Conversely, let v be a valuation of T in I. Let $u_k = vt_k$ for each k in K. Since v is a valuation of T in I, u_k is in I for each k. It is easily seen that $v = \Pi_{Var(T)}\ \rho\Sigma(\underset{k \text{ in } K}{X} \rho_k(u_k))$. Hence v is in $\varphi_T(I)$. □

The second result states the equivalence between expressions and conjunctive queries. (Other types of algebraic expressions have been shown equivalent to conjunctive queries [CM,A]).To do that, we will need two constructions, the concept of normalized expression, and three lemmas.

Definition

An expression φ is <u>normalized</u> if $\varphi = \Pi_V\ \Sigma(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U}))$ for some string Σ of restrictions and some set $\{\rho_k\ /\ k \text{ in } K\}$ of renamings.

In Construction 2, we use a normalized expression without the projection to build a tableau. In Construction 3, we will use a normalized expression to build a conjunctive query.

Construction 2 : From (some particular) expressions to tableaux.

Let $\varphi \equiv \Sigma(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U}))$ be an expression with Σ a string of restrictions. Let $\sim_\varphi$ be the relation over $\underset{k \text{ in } K}{\cup} \rho_k(\mathcal{U})$ defined by $A \sim B$ if $\sigma_{A=B}$ is in Σ . Let $\approx_\varphi$ be the reflexive, symmetric, transitive closure of $\sim_\varphi$. Then $T_\varphi = \{t_k\ /\ k \text{ in } K\}$ is the tableau over $\mathcal{U}$ defined by $t_k(A) = [\rho_k(A)]_{\approx_\varphi}$ for each k in K and A in $\mathcal{U}$. □

We now construct a conjunctive query corresponding to a normalized expression.

Construction 3 : From normalized expression to conjunctive query

Let φ be a normalized expression. Then $\varphi \equiv \Pi_V \varphi_1$ for some φ_1 as in Construction 1. Then $Q_\varphi \equiv < T_{\varphi_1}, t >$ where t is the tuple over V defined by $t(A) = [A]_{\approx_{\varphi_1}}$ for each A in V.

By analogy, T_φ will also denote the tableau T_{φ_1} and $[A]_{\approx_\varphi}$ the equivalence class $[A]_{\approx_{\varphi_1}}$ for each A in $\beta(\varphi_1)$. □

The first lemma states that φ and Q_φ define the same mapping. Its proof is obvious and so omitted.

Lemma 2.1.

For each normalized expression, $\varphi = Q_\varphi$. □

We will prove (Lemma 2.3) that the set of mappings coincides with the set of mappings defined by normalized expression. To do that, we need the following set of rules() :

Definition

Let $\mathfrak{R}$ be the following set of rules :

Projection movers :

(1a) $\sigma_{A=B}\Pi_V = \Pi_V\sigma_{A=B}$

(1b) $\Pi_{\rho(\mathcal{U})}\rho = \rho\Pi_{\mathcal{U}}$

(1c) $\underset{i}{X}\, \Pi_{\mathcal{U}_i}\varphi_i = \Pi_{\underset{i}{\cup}\mathcal{U}_i}\, \underset{i}{X}\, \varphi_i$

() Since cartesian product is commutive and associative, we shall use $\underset{i=1,n}{X} \varphi_i$ as a shortened notation for $\varphi_1 X(\varphi_2 X(\varphi_3 \ldots X\varphi_4)\ldots)$. In the various rules, the expressions are supposed to be well formed. Finally, to simplify the notation, the restrictions of ρ are denoted ρ given that then definition sets are well understood, e.g., in (1b) $\rho\Pi_{\mathcal{U}}$ stands for $\rho|_{\mathcal{U}}\, \Pi_{\mathcal{U}}$

Restriction movers : (2a) $\rho\sigma_{A=B} = \sigma_{\rho(A)=\rho(B)}\ \rho$

(2b) $(\sigma_{A=B}\ \varphi) \times \varphi' = \sigma_{A=B}(\varphi \times \varphi')$

Renaming mover : (3) $\rho \underset{i}{X}\ \varphi_i = \underset{i}{X}\ \rho\varphi_i$

Pseudo-Idempotence : (4a) $\Pi_{\mathcal{U}}\ \Pi_{\mathcal{V}} = \Pi_{\mathcal{U}}$

(4b) If $\rho'' = \rho\rho'$ (as mappings over sets of attributes) then $\rho'' = \rho\rho'$

Cartesian product rules : (5a) $\varphi \times \varphi' = \varphi' \times \varphi$

(5b) $\varphi_1 \times (\varphi_2 \times \varphi_3) = (\varphi_1 \times \varphi_2) \times \varphi_3$

Trivial rule : (0) $\mathcal{U} = \Pi_{\mathcal{U}}\ (\mathcal{U}) = \sigma_{A=A}(\mathcal{U}) = \mathrm{id}(\mathcal{U})$

where id is the identity mapping over $\mathcal{U}$.

The second lemma states the soundness of $\mathcal{R}$.

Lemma( ) 2.2.

For two expressions φ and Ψ, $\varphi \overset{\mathcal{R}}{=} \Psi$ implies $\varphi = \Psi$. □

In the third lemma, $\mathcal{R}$ is used to show how to transform an expression into a normalized expression which defines the same mapping. The proof is only outlined.

Lemma 2.3.

For each expression φ, there exists a normalized expression φ' such that $\varphi \overset{\mathcal{R}}{=} \varphi'$.

( ) $\varphi \overset{\mathcal{R}}{=} \Psi$ means $\varphi = \Psi$ can be proved using the rules in $\mathcal{R}$.

Proof :

In a first step, the projections are moved to the outer level using the projection movers. To do that, it may be necessary to use the trivial rule and rule (1b) to be allowed to apply (1c). Then the projections are combined by (4a).

In a second step, the restrictions are moved to the next outer level using the restriction movers.

Finally, the renamings are moved to the inner lever using the renaming mover and combined using (4b). □

We are now ready to prove that the set of mappings and the set of conjunctive query mappings coincide.

Theorem 2.2.

For each expression φ and conjunctive query Q, there exists a conjunctive query Q_φ and an expression φ_Q such that $\varphi = Q_\varphi$ and $Q = \varphi_Q$.

Proof :

Let φ be an expression. By lemma 2.3, there exists φ' normalized such that $\varphi \overset{\mathcal{R}}{=} \varphi'$. Hence $\varphi = \varphi'$ by lemma 2.2. By lemma 2.1, $\varphi' = Q_{\varphi'}$. Let $Q_\varphi \equiv Q_{\varphi'}$. Then $Q_\varphi = \varphi' = \varphi$.

Let $Q \equiv < T,t >$ be a conjunctive query over $\mathcal{U}$ and $\beta(Q) = V$. (Intuitively, to obtain φ_Q, one just has to rename φ_T and possibly "duplicate" some columns if the same variable appears more than once in the "summary" t). We can assume without loss of generality that $Var(T) \cap V = \emptyset$. Let

$$\varphi_Q \equiv \Pi_V\ \Sigma(\varphi_T \times (\mathop{X}_{B \text{ in } V}\ \rho_B \cdot \Pi_{t(B)}\ \varphi_T))$$

where ρ_B is the renaming over t(B) defined by $\rho(t(B)) = B$ and Σ a string of renamings such that $\{\sigma \text{ in } \Sigma\} = \{\sigma_{B=t(B)}\ /\ B \text{ in } V\}$. Let I be an instance over $\mathcal{U}$ and ω be in Q(I). Then there exists a valuation v of T in I such that $vt = \omega$. By Theorem 2.1, v is in $\varphi_T(I)$. It is easily seen that

$\omega = \Pi_V\ \Sigma(v \times (\underset{B \text{ in } V}{X}\ \rho_B\ \Pi_{t(B)}\ (v)))$. Therefore ω is in

$\Pi_V\ \Sigma(\varphi_T(I) \times (\underset{B \text{ in } V}{X}\ \rho_B\ \Pi_{t(B)}\ \varphi_T(I))) = \varphi_Q(I)$. Thus ω is in $\varphi_Q(I)$.

Hence $Q(I) \subseteq \varphi_Q(I)$.

Conversely, let ω be in $\varphi_Q(I)$. Clearly, $\omega = \Pi_V\ \Sigma(v \times (\underset{B \text{ in } V}{X}\ \rho_B \Pi_{t(B)}\ v))$ for some v in φ_T. By theorem 2.1, v is a valuation of T in I. To conclude the proof, it suffices to remark that ω = vt. Thus ω is in Q(I).
Hence $\varphi_Q(I) \subseteq Q(I)$. □

We conclude this section by presenting two well-known kinds of dependencies, namely, **tgds** and **egds**, and their algebraic counterparts **Atgds** and **Aegds**. First, we present **tgds** and **egds**.

Definition :

Let T, T' be tableaux over $\mathcal{U}$. Then < T,T' > is called a tableau generating dependency (**tgd**). An instance I over $\mathcal{U}$ satisfies < T,T' > iff, for each valuation v of T in I, there exists an extension of v to Var(T) ∪ Var(T') also denoted by v such that vT' ⊆ I.

Definition :

Let T be a tableau over $\mathcal{U}$ and a, b in Var(T). Then < T,a=b > is called an equality generating dependency (**egd**). An instance I over $\mathcal{U}$ satisfies < T,a=b > iff, for each valuation v of T in I, va = vb.

There exist algebraic counterparts to **egd** and **tgd** that we present now.

Definition :

A dependency $\varphi \subseteq \sigma\varphi$ is called an **Aegd**.

Definition :

A dependency $\varphi \subseteq \Psi$ is called an **Atgd** if

(1) $\Psi \equiv \Pi_V \nu$ is normalized, and

(2) $[B]_{\approx_\nu} \cap \beta(\nu) = \{B\}$ for each B in $\beta(\nu)$.

The intuitive meaning of the second condition is that there is no repeated variable in the summary of the corresponding conjunctive query.

The following theorem states the equivalence between (a) **tgds** and **Atgs**, and (b) **egds** and **Aegds**.

Theorem 2.3.

For each **tgd** (resp. **egd**), there exists an equivalent **Atgd** (resp. **Aegd**) and conversely.

Proof :

Each **tgd** is equivalent to an **Atgd**. Let < T,T' > be a **tgd**. Let V = {x / x appears in both T and T'}. It is easily seen that < T,T' > is equivalent to $\Pi_V(\varphi_T) \subseteq \Pi_V(\varphi_{T'})$ and that $\Pi_V(\varphi_T) \subseteq \Pi_V(\varphi_{T'})$ is an **Atgd** (by construction of $\varphi_{T'}$).

Each **egd** is equivalent to an **Aegd**. Let < T,x=y > be an **egd**. It is easily seen that < T,x=y > is equivalent to the **Aegd** $\varphi_T \subseteq \sigma_{x=y}\,\varphi_T$.

Each **Atgd** is equivalent to a **tgd**. Let $\varphi_1 \subseteq \Psi_1$ be an **Atgd**. By lemma 2.3, we can assume without loss of generality that φ_1 and Ψ_1 are normalized. Let $\varphi_1 \equiv \Pi_V\varphi$ and $\Psi_1 = \Pi_V\Psi$. Consider T_φ and T_Ψ . We can also assume without loss of generality that $Var(T_\varphi) \cap Var(T_\Psi) = \emptyset$. Let T be the tableau obtained from T_φ by replacing $[B]_{\approx_\Psi}$ by $[B]_{\approx_\varphi}$ for each B in V. (This is possible since $[B]_{\approx_\Psi} \neq [B']_{\approx_\Psi}$ for each $B \neq B'$ in V by (2) of the definition of **Atgd**). It is easily seen that $\varphi_1 \subseteq \Psi_1$ is equivalent to < T_φ,T > .

Each **Aegd** is equivalent to an **egd**. Let $\varphi \subseteq \sigma_{A=B}\varphi$ be an **Aegd**. By lemma 2.3, we can assume without loss of generality that φ is normalized. Let $\varphi \equiv \Pi_V \Psi$. It is easily seen that $\varphi \subseteq \sigma_{A=B}\varphi$ is equivalent to the egd $< T_\Psi, [A]_{\underset{\Psi}{\approx}} = [B]_{\underset{\Psi}{\approx}} >$. □

III. - CONTAINMENT BETWEEN MAPPINGS

In this section, we present a set of rules which is sound and complete for proving containment between (general) expressions. The proof of the completeness will be based on a well-known result on homomorphism of conjunctive queries [CM]. We first recall the definition of tableau homomorphism.

Definition :

Let T and T' be tableaux over $\mathcal{U}$. Then a homomorphism h from T into T' is a mapping from Var(T) into Var(T') such that $h(T) \subseteq T'$.

The definition is extended to conjunctive queries.

Definition :

Let < T,t > and < T',t' > be two conjunctive queries over $\mathcal{U}$ with $\beta(< T,t >) = \beta(< T',t' >)$. Then a homomorphism h from < T,t > into < T',t' > is a mapping from Var(T) into Var(T') such that $h(T) \subseteq T'$ and $h(t) = t'$.

Now we have :

Theorem 3.1. [CM]

Let Q and Q' be two conjunctive queries. Then $Q \subseteq Q'$ iff there exists a homomorphism from Q' to Q. □

Our next result is concerned with the proof of containment between mappings. We shall need extra rules.

Notation :

Let $\mathcal{R}_2 = \mathcal{R}_1 \cup \{(6),(7),(8),(9)\}$

(6) $\sigma\varphi \subseteq \varphi$

(7) if $\varphi_1 \subseteq \varphi_2$ and $\varphi_3 \subseteq \varphi_4$, then $\Pi\varphi_1 \subseteq \Pi\varphi_2$,

$\sigma\varphi_1 \subseteq \sigma\varphi_2$, $\rho\varphi_1 \subseteq \rho\varphi_2$ and $\varphi_1 \times \varphi_3 \subseteq \varphi_2 \times \varphi_4$.

(8) a/ $\sigma\sigma' = \sigma'\sigma$

b/ $\sigma(\sigma'\sigma'') = (\sigma\sigma')\sigma''$

c/ $\sigma_{A=B}\ \sigma_{B=C} = \sigma_{A=B}\ \sigma_{B=C}\ \sigma_{A=C}$

d/ $\sigma_{A=B} = \sigma_{B=A}$

(9) $\varphi \times \Pi_{\emptyset}(\mathcal{U}) = \varphi$

Now we have :

Theorem 3.2 :

Let φ and Ψ be two expressions. Then $\varphi \subseteq \Psi$ iff $\varphi \overset{\mathcal{R}_2}{\subseteq} \Psi$.

Proof :

It is easily verified that $\mathcal{R}_2$ is sound. Therefore $\varphi \overset{\mathcal{R}_2}{\subseteq} \Psi$ implies $\varphi \subseteq \Psi$. Conversely, suppose that $\varphi \subseteq \Psi$. By lemma 2.1, $\varphi \overset{\mathcal{R}_2}{=} \Pi_V\Sigma(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U}))$ and $\Psi \overset{\mathcal{R}_2}{=} \Pi_V\Sigma'(\underset{k \text{ in } K'}{X} \rho'_k(\mathcal{U}))$ for some $U = \alpha(\varphi)$, $V = \beta(\varphi)$, Σ, Σ', K, K', $\{\rho_k$ / k in $K\}$ and $\{\rho'_k$ / k in $K'\}$. Since $\varphi \subseteq \Psi$, $Q_\varphi \subseteq Q_\Psi$. By Theorem 3.1, there exists a homomorphism from $Q_\Psi \equiv < \{t'_k$ / k in $K'\}, t' >$ into $< \{t_k$ / k in $K\}, t >$. We can assume without loss of generality that $K' \subseteq K$ and $h(t'_k) = t_k$ for each k in K'.

Using (6) and (7), one can obtain a normalized expression $\Psi_1 \equiv \Pi_V \Sigma_1' \Sigma' (\underset{k \text{ in } K'}{X} \rho_k'(\mathcal{U}))$ such that $\Psi_1 \overset{\mathcal{R}_2}{\subseteq} \Psi$ and there exists a one-to-one homomorphism h_1 from $Q_{\Psi_1} = < \{t_k^1 / k \text{ in } K'\}, t^1 >$ into Q_φ . (Σ_1' contains all the restrictions of the form $\sigma_{\rho_i(A) = \rho_j(B)}$ for some i, j, A, B such that $h(t_i'(A)) = h(t_j'(B))$).

Let Σ_1 be a finite string of restrictions such that $\{ <A,B> / \sigma_{A=B} \text{ in } \Sigma_1 \} = \{ <A,B> / A \approx_\varphi B \}$.

By (8),

$$\varphi_1 \equiv \Pi_V \Sigma_1 (\underset{k \text{ in } K}{X} \rho_k(\mathcal{U})) \overset{\mathcal{R}_2}{=} \varphi$$

(Note that there is a one-to-one homomorphism form Q_{Ψ_1} into Q_{φ_1}.)

Let Σ_2 be a finite string of restrictions such that $\{ \sigma \text{ in } \Sigma_2 \} = \{ \sigma_{A=B} \text{ in } \Sigma_1 / AB \subseteq \underset{k \text{ in } K'}{\cup} \rho_k(\mathcal{U}) \}$.

Thus

$$\varphi_1 \overset{\mathcal{R}_2}{\subseteq} \Pi_V \Sigma_2 (\underset{k \text{ in } K}{X} \rho_k(\mathcal{U})) \quad \text{by (6)}$$

$$\varphi_1 \overset{\mathcal{R}_2}{\subseteq} \Pi_V ((\Sigma_2 (\underset{k \text{ in } K'}{X} \rho_k(\mathcal{U}))) \times (\underset{k \text{ in } K-K'}{X} \rho_k(\mathcal{U}))) \quad \text{by (2b)}$$

$$\varphi_1 \overset{\mathcal{R}_2}{\subseteq} \Pi_V \Sigma_2 (\underset{k \text{ in } K'}{X} \rho_k(\mathcal{U})) \times \Pi_\emptyset (\underset{k \text{ in } K-K'}{X} \rho_k(\mathcal{U})) \quad \text{by (1c)}$$

$$\varphi_1 \overset{\mathcal{R}_2}{\subseteq} \Pi_V \Sigma_2 (\underset{k \text{ in } K'}{X} \rho_k(\mathcal{U})) \equiv \varphi_2 \quad \text{by (9)}$$

It is easily seen that there is a one-to-one homomorphism from Q_{Ψ_1} onto Q_{φ_2}.

Intuitively, φ_2 and Ψ_1 differ only by their "dummy" attributes and the ordering of Σ_2 and $\Sigma_1'\Sigma'$. Let h_2 be the isomorphism between Q_{φ_2} and Q_{Ψ_1}. Also let

$$\varphi_2 \equiv \Pi_V \sigma_1 \ldots \sigma_n (\underset{k \text{ in } K}{X} \rho_k(\mathcal{U})) \quad \text{and}$$

$$\Psi_1 \equiv \Pi_V \sigma_1' \ldots \sigma_m' (\underset{k \text{ in } K}{X} \rho_k'(\mathcal{U}))$$

(Recall that $h_2(t'_k) = t_k$ if t_k, t'_k are, respectively the tuples in Q_{φ_2}, Q_{Ψ_1} corresponding to ρ_k, ρ'_k). Let $\{A_k \,/\, A \text{ in } \mathcal{U},\ k \text{ in } K\}$ be a set of new attributes. Let ρ be the renaming over $\bigcup_{k \text{ in } K} \rho_k(\mathcal{U})$ defined by $\rho(\rho_k(A)) = A_k$ for each A and each k. Then

$$\varphi_2 \overset{\mathcal{R}_2}{=} \Pi_V\, \rho^{-1} \rho\, \sigma_1 \dots \sigma_n \left(\mathop{X}_{k \text{ in } K} \rho_k(\mathcal{U}) \right),$$

(a) $$\varphi_2 \overset{\mathcal{R}_2}{=} \Pi_V\, \rho^{-1}\, \bar{\sigma}_1 \dots \bar{\sigma}_n \left(\mathop{X}_{k \text{ in } K} \rho_k(\mathcal{U}) \right)$$

where, for each ℓ, $1 \le \ell \le n$, $\bar{\sigma}_\ell = \sigma_{\rho\rho_i(A) = \rho\rho_j(B)}$ if $\sigma_\ell = \sigma_{\rho_i(A) = \rho_j(B)}$.

Similarly,

(b) $$\Psi_1 \overset{\mathcal{R}_2}{=} \Pi_V\, \rho'^{-1}\, \bar{\sigma}'_1 \dots \bar{\sigma}'_m \left(\mathop{X}_{k \text{ in } K} \rho'\rho'_k(\mathcal{U}) \right)$$

where $\rho'(\rho_k(A)) = A_k$ for each A, each k, and $\bar{\sigma}'_\ell = \sigma_{\rho'\rho_i(A)} = \sigma_{\rho'\rho_j(B)}$ if $\sigma_\ell = \sigma_{\rho_i(A)=\rho_j(B)}$. Clearly, $\rho\rho_k(\mathcal{U}) \overset{\mathcal{R}_2}{=} \rho'\rho'_k(\mathcal{U})$ for each k.

By (8) $\bar{\sigma}_1 \dots \bar{\sigma}_n(W) \overset{\mathcal{R}_2}{=} \bar{\sigma}'_1 \dots \bar{\sigma}'_m(W)$, where $W = \bigcup_{k \text{ in } K} \rho\rho_k(\mathcal{U})$.

Hence

(c) $$\bar{\sigma}_1 \dots \bar{\sigma}_n \left(\mathop{X}_{k \text{ in } K} \rho\rho_k(\mathcal{U}) \right) \overset{\mathcal{R}_2}{=} \bar{\sigma}'_1 \dots \sigma'_m \left(\mathop{X}_{k \text{ in } K} \rho'\rho_k(\mathcal{U}) \right)$$

Let $V' = \{A_k \,/\, A \text{ in } \mathcal{U},\ k \text{ in } K,\ \rho^{-1}(A_k) \text{ in } V\}$. Then it is easily seen that $V' = \{A_k \,/\, A \text{ in } \mathcal{U},\ k \text{ in } K,\ \rho^{-1}(A_k) \text{ in } V\}$ and $\rho^{-1}|_{V'} = \rho'^{-1}|_{V'}$. By (1b)

(d) $$\Pi_V \rho^{-1} \overset{\mathcal{R}_2}{=} \rho^{-1}\, \Pi_{V'} \overset{\mathcal{R}_2}{=} \rho'^{-1}\, \Pi_{V'} \overset{\mathcal{R}_2}{=} \Pi_V \rho'^{-1}$$

Thus $\varphi_2 \overset{\mathcal{R}}{\underset{\mathcal{R}_2}{=}} \Psi_1$ by (a), (b), (c), (d). Therefore $\varphi \overset{\mathcal{R}_2}{=} \varphi_1 \overset{\mathcal{R}_2}{\subseteq} \varphi_2 \overset{\mathcal{R}_2}{=} \Psi_1 \overset{\mathcal{R}_2}{\subseteq} \Psi$.
Hence $\varphi \overset{\mathcal{R}_2}{\subseteq} \Psi$ which concludes the proof. □

IV. - IMPLICATION OF DEPENDENCIES

In this section, we exhibit a sound and complete set of inference rules for dependencies. The set $\mathcal{R}_2$ is not complete. We have to extend it into $\mathcal{R}$.

Definition :

Let $\mathcal{R} = \mathcal{R}_2 \cup \{10,11\}$ where

(10) $\varphi \subseteq \sigma\Psi$ implies $\varphi \subseteq \sigma\varphi$

(11) $\Pi_{AB}\varphi \subseteq \sigma_{A=B}\, \Pi_{AB}\varphi$ implies $\varphi \subseteq \sigma_{A=B}\varphi$.

In order to prove that $\mathcal{R}$ is sound and complete for proving implication of dependencies, we need three lemmas. The first lemma states the soundness of $\mathcal{R}$.

Lemma 4.1. $\mathcal{R}$ is sound.

Proof :

(10) Let $\mathcal{U}$ be a set of attributes, φ and Ψ two expressions over $\mathcal{U}$ with $\beta(\varphi) = \beta(\Psi)$, and A, B two attributes in $\beta(\Psi)$. Let I be an instance over $\mathcal{U}$ such that $\varphi(I) \subseteq \sigma_{A=B}\, \Psi(I)$. Let v be in $\varphi(I)$. Then v is in $\sigma_{A=B}\Psi(I)$. Therefore $v(A) = v(B)$. Since v is in $\varphi(I)$ and $v(A) = v(B)$, v is in $\sigma_{A=B}\varphi(I)$. Thus $\varphi(I) \subseteq \sigma_{A=B}\varphi(I)$. Hence $\varphi \subseteq \sigma_{A=B}\varphi$, so (10) is sound.

(11) Let φ be an expression over $\mathcal{U}$ and $AB \subseteq \beta(\Psi)$. Let I be an instance over $\mathcal{U}$. Let u be in $\varphi(I)$. Then u[AB] is in $\Pi_{AB}\varphi(I) \subseteq \sigma_{A=B}\, \Pi_{AB}\varphi(I)$. Thus u[AB](A) = u[AB](B), i.e., $u(A) = u(B)$. Since u is in $\varphi(I)$ and $u(A) = u(B)$, u is in $\sigma_{A=B}\varphi(I)$. Therefore $\varphi(I) \subseteq \sigma_{A=B}\varphi(I)$. Hence $\varphi \subseteq \sigma_{A=B}\varphi$, so (11) is sound. □

The second lemma states that each dependency is equivalent to a set of **Aegds** and **Atgds**.

<u>Lemma 4.2.</u> For each dependency $\varphi \subseteq \Psi$, there exists a finite set Γ of **Atgds** and **Aegds** s.t. $\varphi \subseteq \Psi \models^{\mathcal{R}} \Gamma$ and $\Gamma \models^{\mathcal{R}} \varphi \subseteq \Psi$.

<u>Proof</u> :

Let $\varphi \subseteq \Psi$ be a dependency. By Lemma 2.3, we can assume that $\varphi \equiv \Pi_V\, \Sigma(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U}))$ and $\Psi \equiv \Pi_V \Sigma'(\underset{k \text{ in } K'}{X} \rho'_k(\mathcal{U}))$. Let $W' = \underset{k \text{ in } K'}{\cup} \rho'_k(\mathcal{U})$. Let $\{e_\ell / \ell \text{ in } L\}$ be the set of equivalence classes of $\approx_\Psi$ ($e_\ell \subseteq W'$) such that $e_\ell \cap V \neq \emptyset$. For each ℓ in L, let B_ℓ be in $e_\ell \cap V$ and $e'_\ell = (e_\ell - V) \cup \{B_\ell\}$. Let Σ'_1 and Σ'_2 be two finite strings of restrictions such that

$$\{\sigma \text{ in } \Sigma'_1\} = \{\sigma_{A=B} \;/\; AB \subseteq [A]_\Psi \text{ and } [A]_\Psi \cap V = \emptyset\}$$

$$\{\sigma_{A=B} \;/\; AB \subseteq e'_\ell\}$$

$$\{\sigma \text{ in } \Sigma'_2\} = \{\sigma_{A=B} \;/\; \ell \text{ in L and } AB \subseteq e_\ell \cap V\}$$

Clearly, $\Sigma'(W) \overset{\mathcal{R}}{=} \Sigma'_1\, \Sigma'_2(W)$ by (8). Thus $\Psi \overset{\mathcal{R}}{=} \Pi_V \Sigma'_1 \Sigma'_2(\underset{k \text{ in } K'}{X} \rho'_k(\mathcal{U}))$. Let $\Psi' \equiv \Pi_V \Sigma'_1(\underset{k \text{ in } K'}{X} \rho'_k(\mathcal{U}))$.

To conclude the proof, it suffices to show

(α) $\varphi \subseteq \Psi \models^{\mathcal{R}} \varphi \subseteq \Psi'$;

(β) $\varphi \subseteq \Psi \models^{\mathcal{R}} \varphi \subseteq \sigma\varphi$ for each σ in Σ'_2 ; and

(γ) $\{\varphi \subseteq \Psi'\} \cup \{\varphi \subseteq \sigma\varphi \;/\; \sigma \text{ in } \Sigma'_2\} \models^{\mathcal{R}} \varphi \subseteq \Psi$.

For suppose (α), (β) and (γ) holds. Clearly, $\varphi \subseteq \Psi'$ is an **Atgd** and $\varphi \subseteq \sigma\varphi$ an Aegd for each σ. Thus, $\varphi \subseteq \Psi \models\dashv \Gamma = \{\varphi \subseteq \Psi'\} \cup \{\varphi \subseteq \sigma\varphi / \sigma \text{ in } \Sigma'_2\}$ for Γ a set of **Atgds** and **Aegds** which concludes the proof.

First consider (1). By (6), $\Sigma'_1\Sigma'_2(W') \overset{\mathcal{R}}{\subseteq} \Sigma'_1(W')$. Thus, $\Psi \overset{\mathcal{R}}{\subseteq} \Psi'$. Therefore $\varphi \subseteq \Psi \models^{\mathcal{R}} \varphi \subseteq \Psi'$.

Consider (2). Let σ be in Σ_2'. By(8), $\Sigma_1'\Sigma_2'(W) \subseteq^{\mathcal{R}} \sigma\Sigma_1'\Sigma_2'(W')$. Since $\Sigma'(W') =^{\mathcal{R}} \Sigma_1'\Sigma_2'(W')$, $\Sigma'(W') \subseteq^{\mathcal{R}} \sigma\Sigma'(W')$. Thus $\Psi \subseteq^{\mathcal{R}} \sigma\Psi$. Since $\Psi \subseteq^{\mathcal{R}} \sigma\Psi$, $\varphi \subseteq \Psi \models^{\mathcal{R}} \varphi \subseteq \sigma\varphi$ by (10).

Finally, consider (3). Let $\Gamma = \{\varphi \subseteq \Psi'\} \cup \{\varphi \subseteq \sigma\varphi / \sigma \text{ in } \Sigma_2'\}$. Let $\sigma_{A=B}$ be in Σ_2' ($AB \subseteq V$). Since $\varphi \subseteq \Psi'$ is in Γ, $\Gamma \models \sigma\varphi \subseteq \sigma\Psi'$ by (7). Since $\varphi \subseteq \sigma\varphi$ is in Γ and $\Gamma \models \sigma\varphi \subseteq \sigma\Psi'$, $\Gamma \models \varphi \subseteq \sigma\Psi'$. By induction on the restrictions in Σ_2', $\Gamma \models \varphi \subseteq \Sigma_2'\Psi'$.

$$\text{By (1), } \Psi' \equiv \Sigma_2' \, \Pi_V \Sigma_1' \Big(\underset{k \text{ in } K'}{X} \rho_k'(\mathcal{U}) \Big) \overset{\mathcal{R}}{=} \Pi_V \Sigma_2' \Sigma_1' \Big(\underset{k \text{ in } K'}{X} \rho_k'(\mathcal{U}) \Big) =^{\mathcal{R}} \Psi.$$

Hence $\Gamma \models \varphi \subseteq \Psi$ which concludes the proof. $\square$

In order to prove that $\mathcal{R}$ is sound and complete we shall simulate the chase process. We recall briefly the chase process. It uses two basic rules.

<u>t-rule</u> - Let T be a tableau and < T',T" > a **tdg**. Then < T',T" > is applicable on T if their exists a homomorphism h from T' to T. The result of applying the rule is the tableau $T \cup h(T")$.

<u>e-rule</u> - Let T be a tableau and < T',a=b > an **egd**. Then < T',a=b > is applicable on T if their exists a homomorphism h from T' to T. The result of applying the rule is the tableau obtained by replacing b by a everywhere in T.

Given a **tgd** $< T,\hat{T} >$ or an **egd** < T,a=b > and a set Γ of **tgds** and **egds**. The chase starts from T and constructs a sequence $T = T_1, T_2, \ldots$ such that T_{i+1} is obtained from T_i by applying a t-rule or an e-rule. The chase process terminates successfully in the case of a **tgd** $< T,\hat{T} >$ if $\hat{T}$ is included in T_{i+1} for some i. It terminates successfully in the case of an **egd** < T,a=b > if an application of an e-rule on T_i (for some i) replaces b by a. It is shown (BV2) that the chase process terminates successfully iff $\Gamma \models < T,\hat{T} >$ in the tgd case or $\Gamma \models$ < T,a=b > in the egd case.

We are ready to prove that $\mathcal{R}$ is complete for proving implication of dependencies.

Lemma 4.3. $\mathcal{R}$ is complete for proving implication of dependencies.

Proof :

Let Γ be a set of dependencies and g a dependency such that $\Gamma \models g$. We will prove that $\Gamma \models^{\mathcal{R}} g$. By Lemma 4.2, we can assume without loss of generality that $\Gamma = \Gamma_1 \cup \Gamma_2$, where Γ_1 is a set of **Atgds** and Γ_2 is a set of **Aegds**, and that g is either an **Atgd** or an **Aegds**. Let $g \equiv \varphi \subseteq \Psi$ be in Γ_1. By theorem 2.3, there exists T and T' such that $g \models \dashv < T,T' >$. Let $V = Var(T) \cap Var(T')$. By inspection of the proof of Theorem 2.3, one can easily see that $\varphi =^{\mathcal{R}} \Pi_V(\varphi_T)$ and $\Psi =^{\mathcal{R}} \Pi_V(\varphi_{T'})$. Therefore $\varphi \subseteq \Psi \models^{\mathcal{R}} \Pi_V(\varphi_T) \subseteq \Pi_V(\varphi_{T'})$ and $\Pi_V(\varphi_T) \subseteq \Pi_V(\varphi_{T'}) \models^{\mathcal{R}} \varphi \subseteq \Psi$.

Hence, we can assume w.ℓ.g. that $\Gamma_1 = \{\Pi_V(\varphi_T) \subseteq \Pi_V(\varphi_{T'}) / < T,T' >$ in $\hat{\Gamma}_1$ and $V = Var(T) \cap Var(T')\}$ for some set $\hat{\Gamma}_1$ of **tgds**. Similarly, we can assume w.ℓ.g. that $\Gamma_2 = \{\varphi_T \subseteq \sigma_{a=b}\varphi_T \ / < T,a=b >$ in $\hat{\Gamma}_2\}$ for some set $\hat{\Gamma}_2$ of **egds**.

By Lemma 2.3, we can also assume that

- either $g \equiv \Pi_V\varphi \subseteq \Pi_V\Psi$ for some $\varphi \equiv \Sigma(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U}))$,

  $\Psi \equiv \Sigma' (\underset{k \text{ in } K'}{X} \rho'_k(\mathcal{U}))$ and is an **Atgd**.

- or $g \equiv \varphi \subseteq \sigma\varphi$ for some $\varphi \equiv \Sigma (\underset{k \text{ in } K}{X} \rho_k(\mathcal{U}))$.

Consider the Tableau T_φ and a chase sequence $T_\varphi = T_1, T_2 \ldots$ with respect to $\hat{\Gamma}_1 \cup \hat{\Gamma}_2$. We construct a sequence $\varphi = \varphi_1, \varphi_2 \ldots$ which satisfies the following property

(a) $\varphi_{i+1} = \varphi_{T_{i+1}}$

(b) $\Gamma_1 \cup \Gamma_2 \models \Pi_{W_{i+1}}(\varphi_i) \subseteq \Pi_{W_{i+1}}(\varphi_{i+1})$

where $W_{i+1} = \beta(\varphi_{i+1}) \cap \beta(\varphi_i)$. Suppose that for some i (a) and (b) are satisfied. We now define φ_{i+1} and show that (a) and (b) are also satisfied. Two cases arise :

(α) T_{i+1} is obtained using the t-rule $< T,T' >$. Thus $< T,T' >$ is in Γ_1, so $\Pi_V\varphi_T \subseteq \Pi_V\varphi_{T'}$ is in Γ_1. Since $< T,T' >$ is applied, there exists a homomorphism h from T into T_i such that $T_{i+1} = T \cup h(T')$. Let $\{\bar{a}$ / a in Var(T) $\cup$ Var(T')$\}$ be a set of new distinct symbols. Let $\bar{T}$ and $\bar{T}'$ be the tableaux obtained from T and T', respectively by replacing a by $\bar{a}$ for each a in Var(T) $\cup$ Var(T'). Let ρ be the mapping over Var(T) $\cup$ Var(T') defined by $\rho(a) = \bar{a}$ for each a. By (1b), $\Pi_{\rho(V)}\rho\varphi_T =^{\mathcal{R}} \rho\Pi_V\varphi_T$. Similarly, $\Pi_{\rho(V)}\rho\varphi_{T'} =^{\mathcal{R}} \rho\Pi_V\varphi_{T'}$. Thus

$$\Gamma \models^{\mathcal{R}} \Pi_V\varphi_T \subseteq \Pi_V\varphi_{T'} \models^{\mathcal{R}} \Pi_{\rho(V)}\rho\varphi_T \subseteq \Pi_{\rho(V)}\rho\varphi_{T'}.$$

Since $\Gamma \models^{\mathcal{R}} \varphi_i \subseteq \varphi_i$ and $\Gamma \models^{\mathcal{R}} \Pi_{\rho(V)}\rho\varphi_T \subseteq \Pi_{\rho(V)}\rho\varphi_{T'}$.

$$\Gamma \models^{\mathcal{R}} (\varphi_i \times \Pi_{\rho(V)}\rho\varphi_t) \subseteq (\varphi_i \times \Pi_{\rho(V)}\rho\varphi_{T'})$$

Now let Σ_1 be a finite set of restrictions such that $\{\sigma$ in $\Sigma_1\}$ = $\{\sigma_{h(a)=\bar{a}}$ / a in $T'\}$. By (6) and (7)

$$(*) \qquad \Gamma \models^{\mathcal{R}} \Sigma_1(\varphi_i \times \Pi_{\rho(V)}\rho\varphi_T) \subseteq \Sigma_1(\varphi_i \times \Pi_{\rho(V)}\rho\varphi_{T'})$$

Let $\varphi_{i+1} = \Sigma_1(\varphi_i \times \rho\varphi_{T'})$. It is easily seen that $\varphi_{i+1} = \varphi_{T_{i+1}}$. (Intuitively, the rows in T_i correspond to φ_i, the rows in h(T') correspond to $\rho\varphi_{T'}$ and the matching between variables in T_i and h(T') is simulated by Σ_1). By Theorem 3.2, $\varphi_{i+1} =^{\mathcal{R}} \varphi_{T_{i+1}}$. Thus (a) is true for i+1.

Let $\varphi_i' \equiv \Sigma_1(\varphi_i \times \Pi_{\rho(V)}\rho\varphi_T)$. Let $Q_i \equiv < T_{\varphi_i}, t_i >$ and $Q_i' \equiv < T_{\varphi_i'}, t_i' >$ where t_i, t_i' are the tuples over $\beta(\varphi_i)$ defined by $t_i(a) = [a]_{\approx_{\varphi_i}}$ and $t_i'(a) = [a]_{\approx_{\varphi_i'}}$ for each a in $\beta(\varphi_i)$. By inspection of Q_i and Q_i', one can verify that Q_i, Q_i' are isomorphic. Thus $Q_i = Q_i'$. Clearly, $Q_i = \varphi_i$ and $Q_i' = \Pi_{\beta(\varphi_i)}\varphi_i'$. Also, $W_{i+1} = \beta(\varphi_i) \cap \beta(\varphi_{i+1}) = (\beta(\varphi_i) \cup \rho(V)) \cap \beta(\varphi_i) = \beta(\varphi_i)$ since $\bar{a}$ was a new attribute for each a. Hence $\Pi_{W_{i+1}}(\varphi_i) = Q_i = Q_i' = \Pi_{W_{i+1}}(\varphi_i')$. By Theorem 3.2,

$\Pi_{W_{i+1}}(\varphi_i) \overset{\mathcal{R}}{=} \Pi_{W_{i+1}}(\varphi_i^!)$. Hence

$$\Gamma \models^{\mathcal{R}} \Pi_{W_{i+1}}(\varphi_i) \subseteq \Pi_{W_{i+1}}(\Sigma_1(\varphi_i \times \Pi_{\rho(V)}\rho\varphi_{T'})) \quad \text{by } (*) \ ,$$

$$\Gamma \models^{\mathcal{R}} \Pi_{W_{i+1}}(\varphi_i) \subseteq \Pi_{W_{i+1}}\Pi_{\rho(V)\,\cup\, W_{i+1}}(\Sigma_1(\varphi_i \times \rho\varphi_{T'})) \quad \text{by (1c)},$$

$$\Gamma \models^{\mathcal{R}} \Pi_{W_{i+1}}(\varphi_i) \subseteq \Pi_{W_{i+1}}(\varphi_{i+1}) \quad \text{by (4a)} \ .$$

Therefore (b) is true for i+1. Thus (a) and (b) hold.

(β) T_{i+1} is obtained by the e-rule $< T,a=b >$. Thus $< T,a=b >$ is in $\bar{\Gamma}_2$, so $\varphi_T \subseteq \sigma_{a=b}\varphi_T$ is in Γ_2. Since the rule is applicable, there exists a homomorphism h from T into T_i. Let ρ be the mapping over ab defined by $\rho = h_{|ab}$. Let Q and $Q_i^!$ be the conjunctive queries defined by $\alpha(Q) = \alpha(Q_i^!) = \mathcal{U}$, $\beta(Q) = \beta(Q_i^!) = h(a)h(b)$, $Q \equiv < T,t >$ and $Q' = < T_i,t' >$ where $t(h(a)) = t'(h(a)) = a$ and $(th(b)) = t'(h(b)) = b$. Clearly, $Q = \rho\Pi_{ab}\varphi_T$ and $Q_i^! = \Pi_{h(a)h(b)}\varphi_i$. Also, h is a homomorphism from Q into $Q_i^!$. Therefore $\rho\Pi_{ab}\varphi_T = Q \supseteq Q_i^! = \Pi_{h(a)h(b)}\varphi_i$. By Theorem 3.2, (+) $\Pi_{h(a)h(b)}\varphi_i \overset{\mathcal{R}}{\subseteq} \rho\Pi_{ab}\varphi_T$. Thus

$$\Gamma \models^{\mathcal{R}} \varphi_T \subseteq \sigma_{a=b}\varphi_T \models^{\mathcal{R}} \rho\Pi_{ab}\varphi_T \subseteq \rho\Pi_{ab}\sigma_{a=b}\varphi_T \ ,$$

$$\Gamma \models^{\mathcal{R}} \rho\Pi_{ab}\varphi_T \subseteq \sigma_{(a)=h(b)}\rho\Pi_{ab}\varphi_T \ ,$$

$$\Gamma \models^{\mathcal{R}} \Pi_{h(a)h(b)}\varphi_i \subseteq \sigma_{h(a)=h(b)}\rho\Pi_{ab}\varphi_T \quad \text{by (+)} \ ,$$

$$\Gamma \models^{\mathcal{R}} \Pi_{h(a)h(b)}\varphi_i \subseteq \sigma_{h(a)=h(b)}\Pi_{h(a)h(b)}\varphi_i \quad \text{by (10)} \ ,$$

$$\Gamma \models^{\mathcal{R}} \varphi_i \subseteq \sigma_{h(a)=h(b)}\varphi_i \quad \text{by (11)}.$$

Now let $\varphi_{i+1} \equiv \Pi_{W_{i+1}}\sigma_{h(a)h(b)}\varphi_i$ where $W_{i+1} = \beta(\varphi_i)-h(b)$. (Note that $W_{i+1} = \beta(\varphi_i) \cap \beta(\varphi_{i+1})$). It is easily seen that $\varphi_{i+1} = \varphi_{T_{i+1}}$, i.e. (a) holds. Since $\Gamma \models^{\mathcal{R}} \varphi_i \subseteq \sigma_{h(a)=h(b)}\varphi_i$, $\Gamma \models^{\mathcal{R}} \Pi_{W_{i+1}}(\varphi_i) \subseteq \Pi_{W_{i+1}}\sigma_{h(a)=h(b)}\varphi_i = \Pi_{W_{i+1}}(\varphi_{i+1})$. Thus (b) holds.

In each case ((α) and (β)), (a) and (b) hold. Hence (a) and (b) hold for each i.

We are now ready to prove the lemma. Two cases arise :

(γ) g is an Atgd. Hence $g \equiv \Pi_V(\varphi) \subseteq \Pi_V(\Psi)$.
The chase was concerned with $< T_\varphi, T_\Psi >$. The chase terminates by $T_\Psi \subseteq T_n$ for some n. Consider the conjunctive queries $Q_1 \equiv < T_\Psi, t >$ and $Q_2 \equiv < T_n, t >$ where t is the tuple over $\beta(\Psi)$ defined by $t(a) = [a]_\Psi$ for each a in $\beta(\Psi)$. (Since $T_\Psi \subseteq T_n$, $< T_n, t >$ is a well-defined conjunctive query). Since $T_\Psi \subseteq T_n$, the identity h on Var(Ψ) is a homomorphism from Q_1 into Q_2. Thus $Q_2 \subseteq Q_1$. Clearly, $Q_1 = \Psi$ and $Q_2 = \Pi_{\beta(\Psi)}\varphi_{T_n} = \Pi_{\beta(\Psi)}\varphi_n$. Therefore $\Pi_{\beta(\Psi)}\varphi_n \subseteq \Psi$. By (a) and (b), and since the attributes introduced in $\varphi_1, \ldots \varphi_n$ were new, $W_{n+1} \subseteq W_n \ldots \subseteq W_1$. For the same reasons $V = \beta(\varphi) \cap \beta(\Psi) = \beta(\Psi) \cap W_{n+1}$. Hence, $V \cap \beta(\varphi) \cap \beta(\Psi) \cap W_1 \cap \ldots \cap W_{n+1}$. Thus

$$\Gamma \models^{\mathcal{R}} \Pi_V(\varphi) = \Pi_V(\varphi_1) \subseteq \Pi_V(\varphi_2) \ldots \Pi_V(\varphi_n), \text{ and}$$

$$\Gamma \models^{\mathcal{R}} \Pi_V(\varphi_n) \subseteq \Pi_V(\Psi).$$

Therefore $\quad \Gamma \models^{\mathcal{R}} \Pi_V(\varphi) \subseteq \Pi_V(\Psi)$, i.e., $\Gamma \models^{\mathcal{R}} g$.

(δ) g is an Aegd. Thus $g \equiv < T_\varphi, a=b >$. The last step of the chase equates a and b. That is, $< T_\varphi, a'=b' >$ is applied and there is a homomorphism h from T_φ into T_n such that $h(a') = a$ and $h(b') = b$. Therefore (as seen in (β)) $\Gamma \models^{\mathcal{R}} \varphi_{n-1} \subseteq \sigma_{a=b}\varphi_{n-1}$. Thus $\Gamma \models^{\mathcal{R}} \Pi_{ab}\varphi_{n-1} \subseteq \Pi_{ab}\sigma_{a=b}\varphi_{n-1}$. Since $\Gamma \models^{\mathcal{R}} \Pi_{W_n}(\varphi_1) \subseteq \ldots \subseteq \Pi_{W_n}(\varphi_{n-1})$, $\Gamma \models^{\mathcal{R}} \Pi_{ab}(\varphi_1) \subseteq \ldots \Pi_{ab}\varphi_{n-1}$. Hence $\Gamma \models^{\mathcal{R}} \Pi_{ab}(\varphi) \subseteq \Pi_{ab}\sigma_{a=b}(\varphi_{n-1}$; Therefore

$$\Gamma \models^{\mathcal{R}} \left\{ \begin{array}{l} \varphi \subseteq \varphi \\ \\ \varphi \subseteq \sigma_{a=b}\Pi_{ab}\varphi_{n-1} \end{array} \right\} \models^{\mathcal{R}} \varphi = \sigma_{a=b}\varphi \qquad \text{by (10).}$$

Thus $\Gamma \models^{\mathcal{R}} g$.

In each case, $\Gamma \models^{\mathcal{R}} g$ which concludes the proof. □

Now we can state :

Theorem 4.1.

$\mathcal{R}$ is sound and complete for proving implication of dependencies. □

V. - TYPED VS. NONTYPED

Section 2, 3 and 4 were dealing with algebraic mappings and dependencies. Some particular algebraic mappings and dependencies were studied in [YP], that is the "typed" ones. The distinction between typed and non-typed is now presented. The notion of "weak" typing is then introduced, and a characterization for weakly typed mappingsgiven. Finally, a strong connection between typed and monotone is exhibited.

We start with the definition of typing.

Definition

A typing $\mathcal{C}$ for an expression φ is an equivalence relation over $\mathcal{A}$ such that

(α)	$[A] \neq [B]$	for each A, B in $\alpha(\varphi)$, $A \neq B$;
(β)	$[\rho(A)] = [A]$	for each renaming ρ used in φ and each A in $\alpha(\rho)$; and
(γ)	$[A] = [B]$	for each restriction $\sigma_{A=B}$ used in φ.

Now we have :

Definition

An expression φ is typed if there exists a typing $\mathcal{T}$ for an expression Ψ such that $\varphi = \Psi$. A dependency $\varphi \subseteq \Psi$ is typed if there exists a typing $\mathcal{T}$ for two expressions φ_1 and Ψ_1 such that $\varphi = \varphi_1$ and $\Psi = \Psi_1$.

It turns out that the set of typed mappings coincides roughly with the set of "extended projection join mappings" defined in [YP]. Indeed, typed dependencies coincides with Yannakakis and Papadimitriou's algebraic dependencies.

Theorem 5.1.

The set of typed dependencies coincides with (1) the set of (YP) - algebraic dependencies and (2) the set of embedded implicational dependencies. □

There are intuitively two reasons for a mapping not to be typed : (1) the mapping involves some comparison of entries from different columns of the input relation and (2) the input and output relations have a common attribute , say A, and an entry from column A in the output relation can come from another column, say B≠A, in the input relation. The following two examples, Examples 4.1 and 4.2, resp., illustrate (1) and (2), resp.

Example 4.1.

Let $\mathcal{U} = AB$. Then $\varphi_1 \equiv \sigma_{A=B}(\mathcal{U})$ is not typed.

Exemple 4.2.

Let $\mathcal{U} = AB$ and ρ be the renaming over AB defined by $\rho(A)=B$ and $\rho(B) = A$. Then $\varphi_2 \equiv \rho(\mathcal{U})$ is not typed.

This suggests the following definition of "weak" typing.

Definition

A mapping φ is "weakly" typed if there exist an expression φ_1, a typing $\mathcal{C}$ for φ_1, and a renaming ρ such that $\varphi = \rho\varphi_1$.

It is easily seen that φ_2 is weakly typed whereas φ_1 is not.

Our next result is a characterization of weakly typed mappings. To prove it, we need two lemmas.

Lemma 5.1.

Let φ be an expression of typing $\mathcal{C}$. Then, for each A in $\beta(\varphi)$, $[A] \cap \alpha(\varphi)$ is a set of exactly one attribute.

Proof :

By (α), $\#([A] \cap \alpha(\varphi)) \leq 1$. By (β), $\#([A] \cap \alpha(\varphi)) \geq 1$. □

The second lemma is concerned with the application of a typed expression to a single tuple.

Lemma 5.2.

Let φ be an expression of typing $\mathcal{C}$ and u a tuple over $\alpha(\varphi)$. Then $\varphi(u) = \{v\}$ where v is the tuple over $\beta(\varphi)$ defined by : for each A in $\beta(\varphi)$ $v(A) = u(B)$ when $B = [A] \cap \alpha(\varphi)$.

Proof :

The proof is straighforward (induction on the depth of φ), and so omitted. □

We are now ready to characterize weakly typed mappings.

Theorem 5.2.

Let φ be an expression over $\mathcal{U}$. Then the following three conditions are equivalent

(1) φ is weakly typed ;

(2) $\varphi(u) \neq \emptyset$ for all tuples u in Tup($\mathcal{U}$) ; and

(3) $\varphi(u) \neq \emptyset$ for some tuple u in Tup($\mathcal{U}$) such that $u(A) \neq u(B)$ for each A and B in $\mathcal{U}$, $A \neq B$.

Proof :

(1) => (2). Since φ is weakly typed, there exist a typed expression φ_1 and a renaming ρ such that $\varphi = \rho\varphi_1$. Let u be in Tup($\mathcal{U}$). Since φ_1 is typed, $\varphi_1(u) \neq \emptyset$ by Lemma 5.2. Thus $\varphi(u) = \rho\varphi_1(u) \neq \emptyset$.

(2) => (3). Obvious.

(3) => (1). Let u be in Tup($\mathcal{U}$) such that $u(A) \neq u(B)$ for each A, B in $\mathcal{U}$, $A \neq B$, and $\varphi(u) \neq \emptyset$. By Lemma 2.3, we can assume without loss of generality that $\varphi = \Pi_V\Sigma(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U}))$. Let $\{A_k \ / \ k \text{ in } K, A \text{ in } \mathcal{U}\}$ be a set of new distinct attributes and ρ be the renaming over $\underset{k \text{ in } K}{U} \rho_k(\mathcal{U})$ defined by $\rho(\rho_k(A)) = A_k$ for each k in K and A in $\mathcal{U}$. Then

$$(*) \qquad \Pi_{\rho(V)}\rho\Sigma(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U})) = \rho\Pi_V(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U})) = \rho\varphi$$

Also

$$(**) \qquad \Pi\rho_{(V)}\rho\Sigma(\underset{k \text{ in } K}{X} \rho_k(\mathcal{U})) = \Pi_{\rho(V)}\Sigma'(\underset{k \text{ in } K}{X} \rho'_k(\mathcal{U})) ,$$

where Σ' is a finite string of renaming such that

$$\{\sigma' \text{ in } \Sigma'\} = \{\sigma_{\rho(A)=\rho(B)} \ / \ \sigma_{A=B} \text{ in } \Sigma\} \text{ and } \rho'_k = \rho\rho_k$$

for each k. By (*) and (**), $\rho\varphi = \Pi_{\rho(V)}\Sigma'(\underset{k \text{ in } K}{X} \rho'_k(\mathcal{U}))$.

To conclude the proof, it suffices to show that (+) there exists a typing for $\Psi \equiv \Pi_{\rho(V)}\ \Sigma'(\underset{k \text{ in } K}{X}\ \rho'_k(\mathcal{U}))$. For suppose (+) holds. Then $\varphi = \rho^{-1}\rho\varphi = \rho^{-1}\Psi$. Hence φ is weakly typed since Ψ has a typing.

To prove (+), let $\mathcal{C}$ be the equivalence relation on $\mathcal{A}$ defined by $[A] = [A_k] = \{A\} \cup \{A_\ell \,/\, \ell \text{ in } K\}$ for each A in $\mathcal{U}$ and k in K, and $[B] = \{B\}$ otherwise. By construction $\mathcal{C}$ satisfies (α) and (β) for Ψ. Suppose that $\mathcal{C}$ doesn't satisfy (γ) for Ψ. Hence, there exist A, B, $A \neq B$, i and j such that $\sigma_{A_i=B_j}$ is in Σ'. Since $(\underset{k \text{ in } K}{X}\ \rho'_k(\mathcal{U}))(A_i) = u(A) \neq u(B) = (\underset{k \text{ in } K}{X}\ \rho'_k(\mathcal{U}))(B_j)$ $\Psi(u) = \emptyset$. Therefore, $\varphi(u) = \rho^{-1}\Psi(u) = \emptyset$, a contradiction. Hence $\mathcal{C}$ satisfies (γ) for Ψ. Thus $\mathcal{C}$ is a typing for Ψ, so (+) holds which concludes the proof. □

As mentionned in the introduction, we now exhibit a connection between typed and monotone. We first define monotone.

Definition

An expression φ such that $\alpha(\varphi) = \beta(\varphi)$ is monotone iff $\varphi(I) \subseteq I$ for each I in Inst($\alpha(\varphi)$).

Now we have :

Theorem 5.3.

Let φ be an expression such that $\alpha(\varphi) = \beta(\varphi)$. Then φ is typed iff φ is monotone.

Proof :

First suppose that φ is typed. It easily follows from Lemma 5.2 that $\varphi(u) = u$ for each u in Tup($\alpha(\varphi)$). Hence $\varphi(I) \supseteq \{\varphi(u)/u \text{ in } I\} = \{u \,/\, u \text{ in } I\} = I$. Thus φ is monotone.

Now suppose φ is monotone. Let u be a tuple in Tup($\mathcal{U}$) such that u(A) $\neq$ u(B) for each A and B in $\mathcal{U}$, A$\neq$B. Since φ is monotone u is in φ(u), so φ(u) $\neq \emptyset$. By Theorem 5.2, there exist an expression φ_1, a typing $\mathcal{C}$ for φ_1 and a renaming ρ such that $\varphi = \rho\varphi_1$. Since φ_1 is typed, φ_1(u)=u by Lemma 5.2. Since u is in φ(u), φ(u) = u. Hence ρ(u) = $\rho\varphi_1$(u) = $\rho\varphi$(u) = u. It is easily seen that ρ(u) = u implies that ρ(A) = A for each A in $\mathcal{U}$, i.e. ρ is the identity renaming. Thus $\varphi = \varphi_1$ and φ_1 has typing $\mathcal{C}$, so φ is typed. □

Corollary 5.1.

Let φ be an expression such that $\alpha(\varphi) = \beta(\varphi)$. Then φ is weakly typed iff $\rho\varphi$ is monotone for some renaming ρ. □

REFERENCES

[A] Abiteboul, S., "Matching Functions and Disaggregations in Databases", Ph.D. Thesis, University of Southern California, 1982.

[ABU] Aho, A.V., C. Beeri, and J.D. Ullman, "The Theory of Joins in Relational Databases", ACM TODS, Vol 4 (1979), pp. 297-314.

[ASU] Aho, A.V., Y. Sagiv, and J.D. Ullman, "Equivalence Among Relational Expressions", SIAM Journal of Computing, Vol. 8, N° 2, May 1979, pp. 218-246.

[Ar] Armstrong, W.W., "Dependency Structures of Database Relationship", Proc. IFIP 74, North Holland (1974), pp. 580-583.

[BFH] Beeri, C., R. Fagin and J.H. Howard, "A Complete Axiomatization for Functional and Multivalued Dependencies in Database Relations", ACM SIGMOD International Symp. on Manag. of Data (1977), pp. 47-61.

[BMSU] Beeri, C., A.C. Mendelson, Y. Sagiv, and J.D. Ullman, "Equivalence of Relational Database Schemes", Proc. 1979 ACM Symp. on the Theory of Computing, pp. 319-329.

[BV] Beeri, C., and M.Y. Vardi, "The Implication Problem for Data Dependencies", Department of Computer Science, The Hebrew University of Jerusalem, May 1980.

[BV1] Beeri, C., and M.Y. Vardi, "A Proof Procedure for Data Dependencies", Tech. Rep., Dept of Comp. Sci., The Hebrew University of Jerusalem, Dec. 1980.

[CH] Chandra, A.K., and D. Harel, "Structure and Complexity of Relational Queries," 21st IEEE Symp. on Found. of Comp. Science, Syracuse, NY (Oct 1980), pp. 333-347.

[CM] Chandra, A.K., and P.M. Merlin, "Optimal Implementation of Conjunctive Queries in Relational Data Bases", Proc. of the 9th annual ACM Symp. on Theory of Comp. (May 1977), pp. 77-90.

[MMS] Maier, D., A.O. Mendelson, and Y. Sagiv, "Testing Implications of Data Dependencies", ACM TODS, 4:4 (Dec 1979), pp. 455-469.

[SU] Sadri, F., and J.D. Ullman, "A Complete Axiomatization for a Large Class of Dependencies on Relational Databases", Proc. of 12th ACM Symp. on Theory of Comp. (Apr. 1980), pp. 117-122.

[YP] Yannakakis, and C. Papadimitriou, "Algebraic Dependencies", Proc. 21st IEEE Symp. on Found. of Comp. Science (1980), pp. 328-332.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique